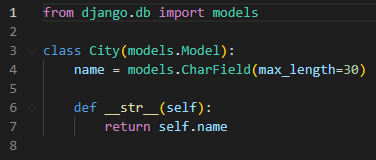
Отчёт по проекту “Приложение погоды”

Реализуемый функционал:

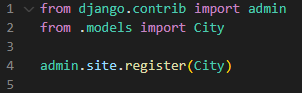
1. Кнопка вывода информации о погоде на текущий момент в выбранном городе
2. Кнопка закрепления информации о погоде в выбранном городе (“Добавить в избранное”)
3. Кнопка удаления закрепленной информации о погоде в выбранном городе (“Удалить из избранного”)
4. Получение прогноза погоды на 5 дней в определенном городе

**Код файла models:**



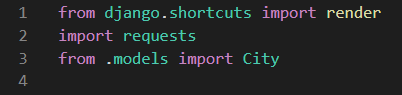
**Рис.1 Импортирование моделей из Django, создание модели City для дальнейшего сохранения городов в ней.**

**Код файла admin:**



**Рис.2 Импортирование модели City и регистрирование её.**

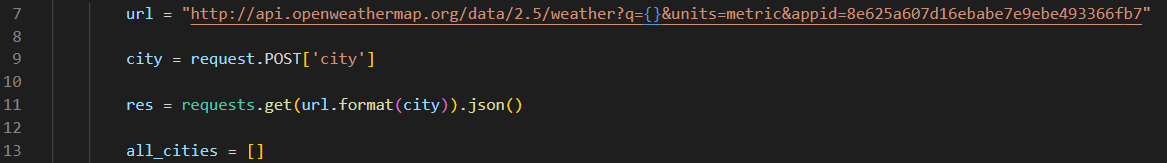
**Код файла views:**



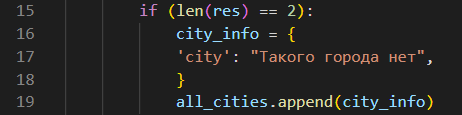
**Рис.3 Подключение библиотеки Requests, импортирование модели City, импортирование функции render.**



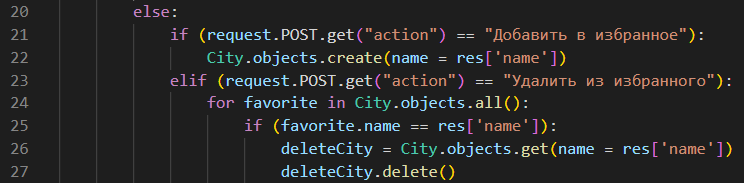
**Рис.4 Создание функции index, проверка метода передачи.**



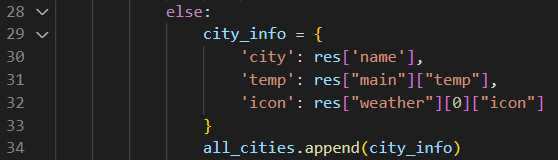
**Рис.5 Добавление url-ссылки для получения информации о погоде в городе, получение имени города и получение информации о городе, путем подставления в url-ссылку названия города, создание массива all\_cities.**



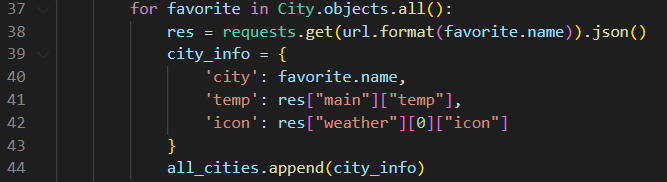
**Рис.6 Проверка на существование города.**



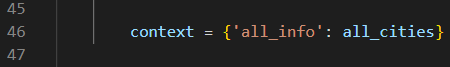
**Рис.7 Добавление и удаление города в модель City (Избранное).**



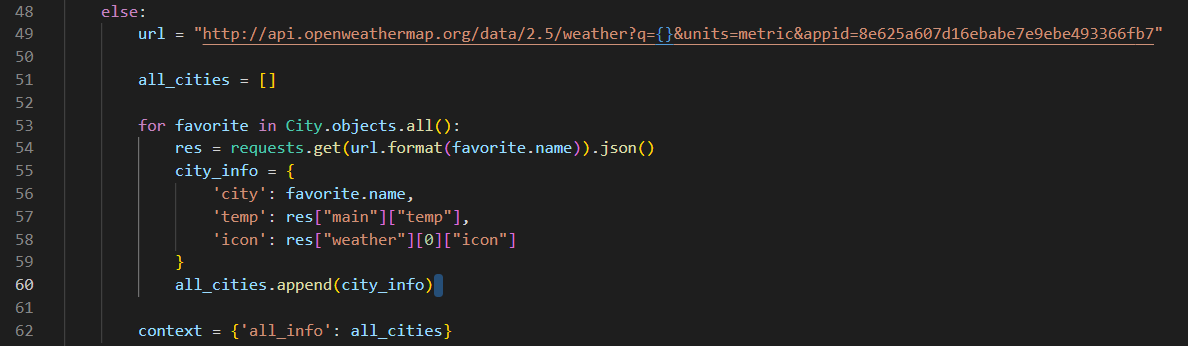
**Рис.8 Добавление ключевой информации в массив all\_cities.**



**Рис.9 Получение информации о всех городах в модели City и дальнейшее добавление информации в массив all\_cities.**



**Рис.10 Добавление массива all\_cities в словарь context.**



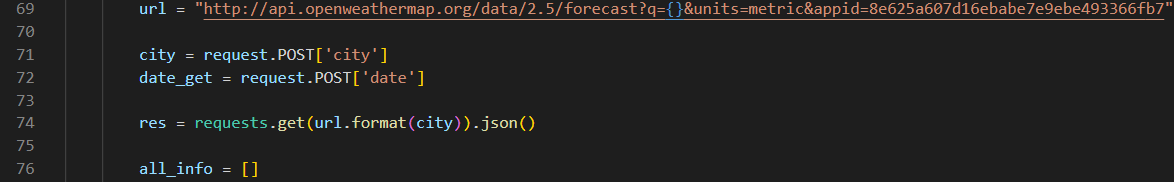
**Рис.11 Если метод POST не был использован, тогда идет передача информации только о городах в модели City.**



**Рис.12 Передача словаря context в html код сайта.**



**Рис.13 Создание функции index, проверка метода передачи.**



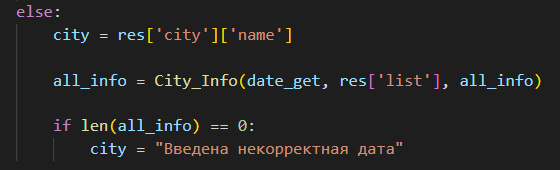
**Рис.14 Добавление url-ссылки для получения информации о прогнозе погоды в городе, получение имени города, даты и получение информации о городе, путем подставления в url-ссылку названия города, создание массива all\_info.**



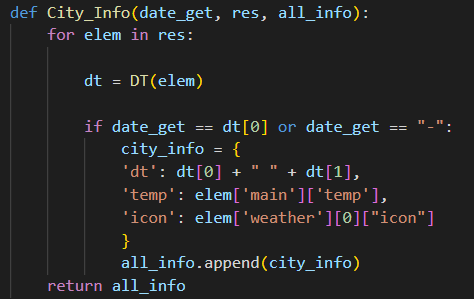
**Рис.15 Проверка на существование города.**



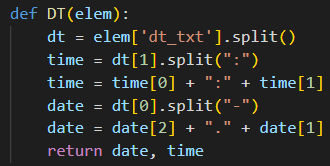
**Рис.16 Проверка на корректность даты.**



**Рис.19 Добавление ключевой информации в массив all\_info и переменную city.**

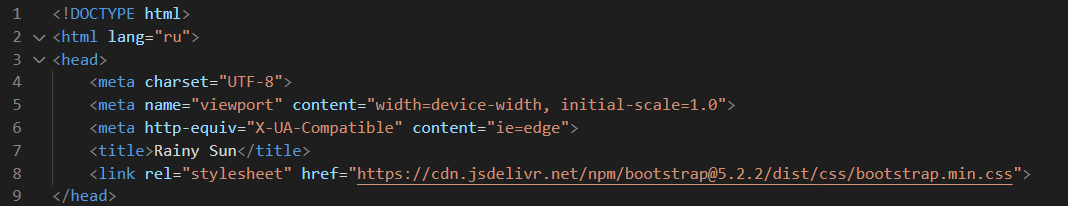


**Рис.20 Создание функции City\_info и выполнение основных представлений.**

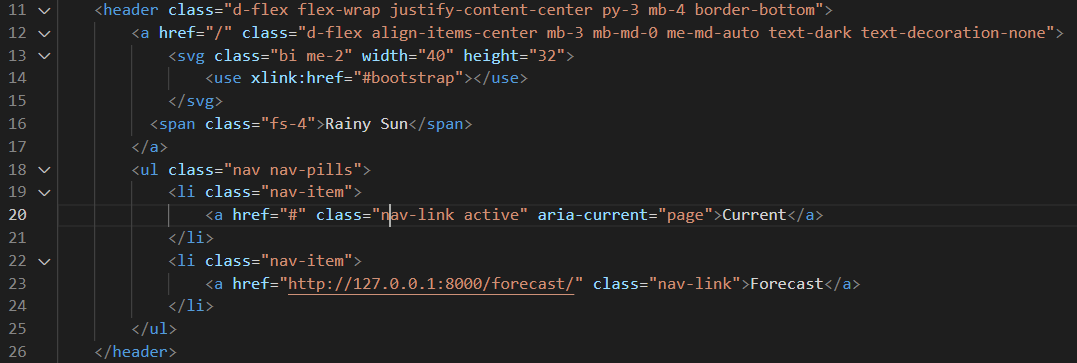


**Рис.21 Создание функции DT и работа с датой и временем в ней.**

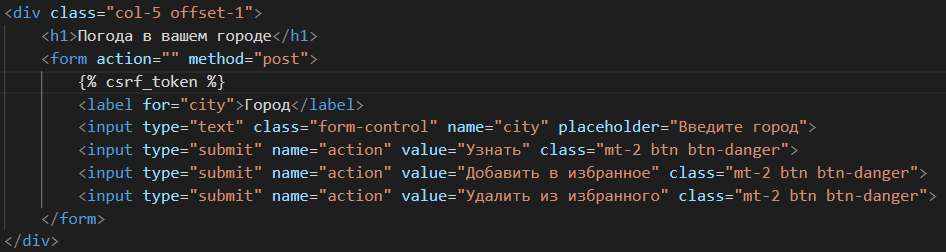
**Код из index.html:**



**Рис.22 Основные настройки сайта и подключение bootstrap.**



**Рис.23 Создание шапки сайта**

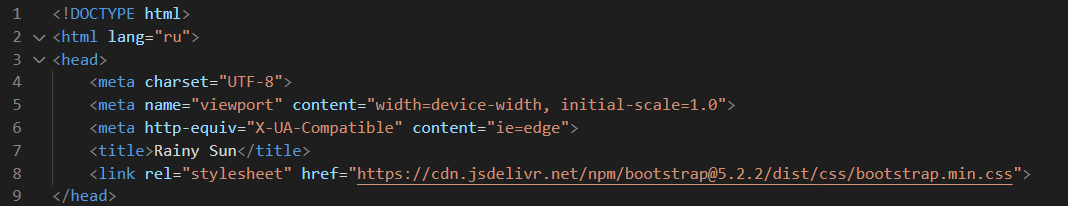


**Рис.24 Создание блока с текстовой строкой и кнопками, подключение csrf\_token.**

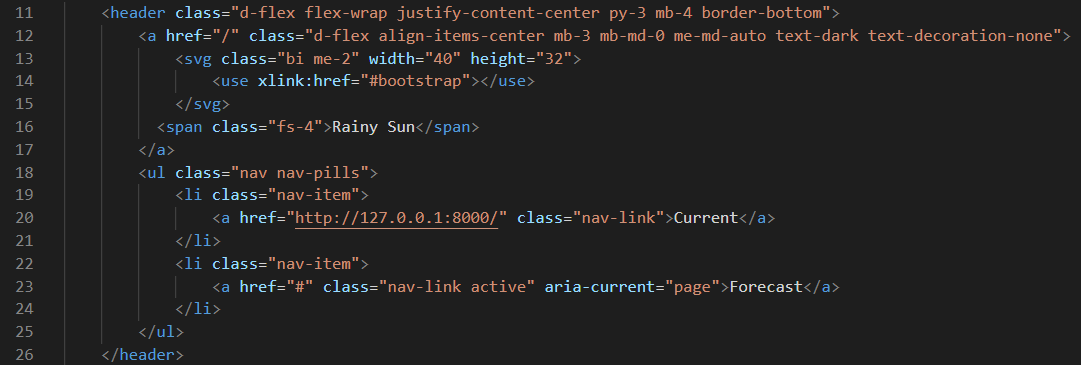


**Рис.25 Создание блока с информацией о погоде в городе, использование цикла для вывода информации об N множестве городов.**

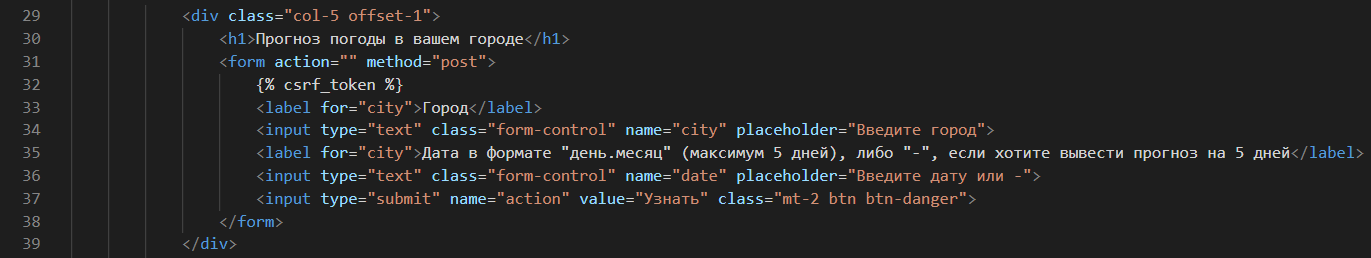
**Код из prognoz.html:**



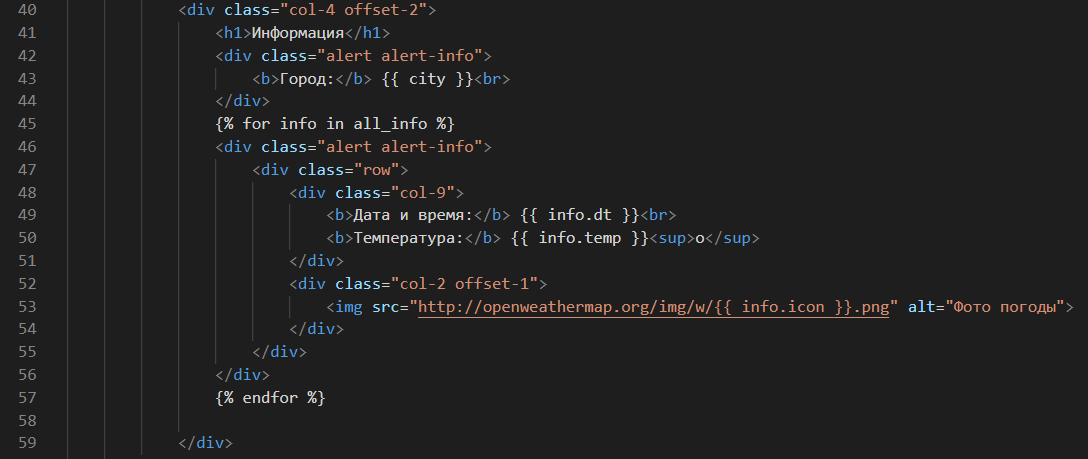
**Рис.26 Основные настройки сайта и подключение bootstrap.**



**Рис.27 Создание шапки сайта**

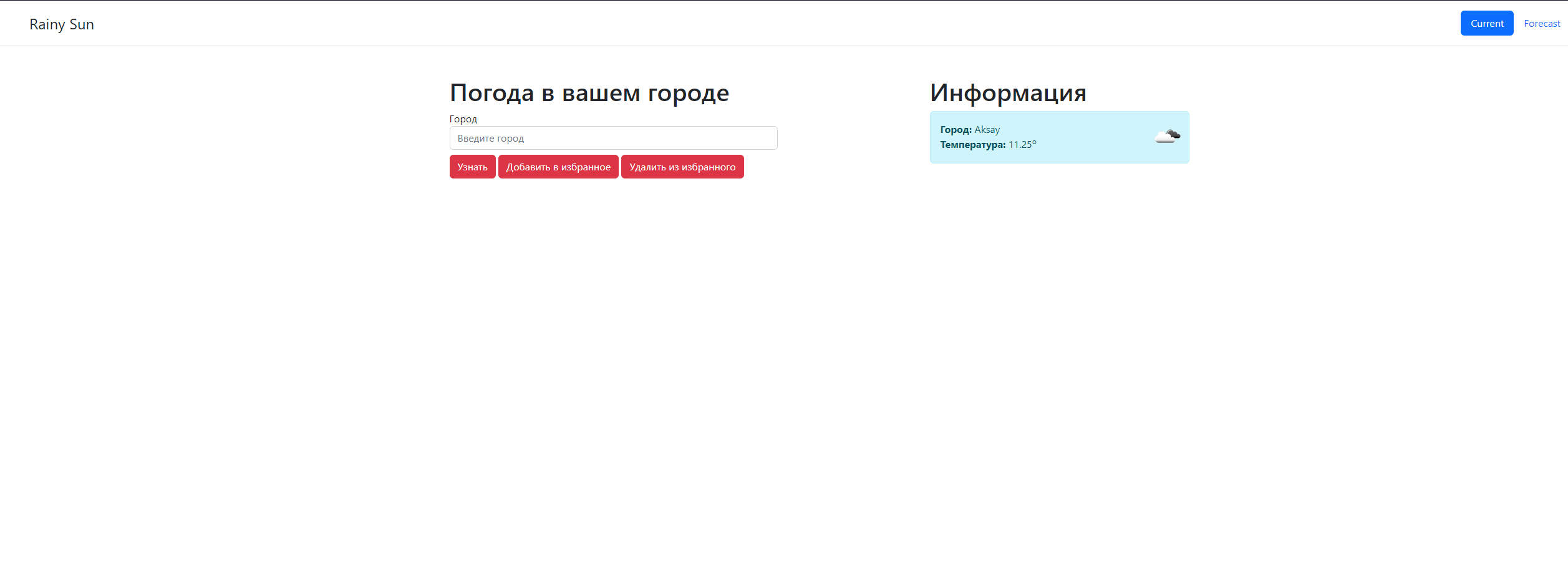


**Рис.28 Создание блока с текстовыми строками и кнопкой, подключение csrf\_token.**

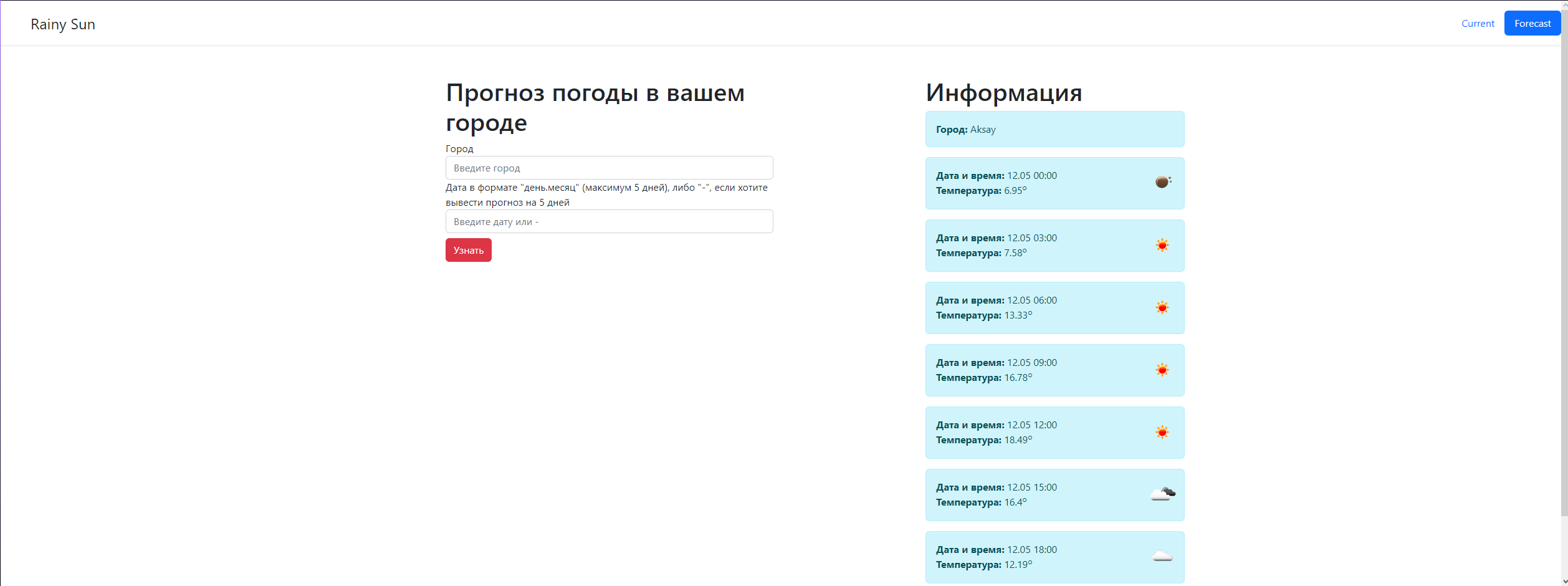


**Рис.29 Создание блока с информацией о прогнозе погоды в городе, использование цикла для вывода информации об N погодном изменении.**

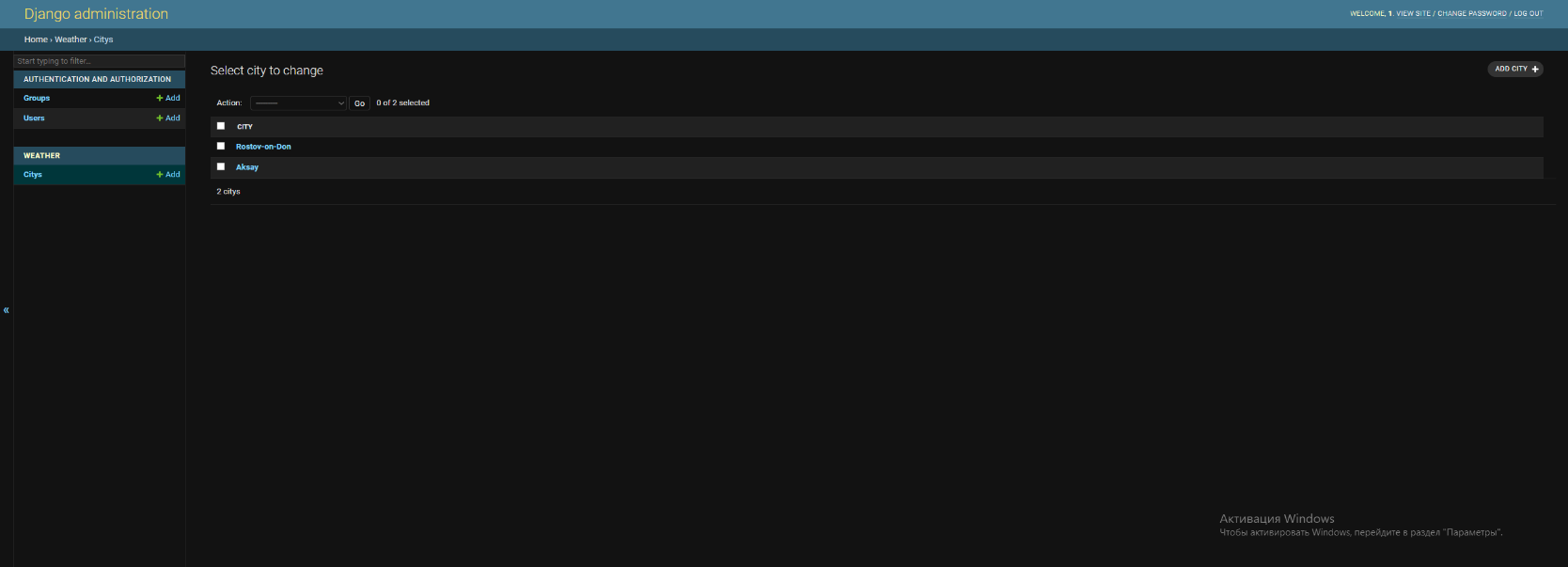
**Скриншоты сайта:**



**Рис.30 Страница получения информации о текущей погоде в городах.**



**Рис.31 Страница получения информации о прогнозе погоды в городе.**

****

**Рис.32 Модель City в Django administration.**